Self-supervised texture segmentation using complementary types of features

Jiebo Luo
Andreas Savakis

Follow this and additional works at: http://scholarworks.rit.edu/article

Recommended Citation
Pattern Recognition 34N11 (2001) 2071-2082

This Article is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Articles by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Self-supervised texture segmentation using complementary types of features

Jiebo Luo\textsuperscript{a} and A.E. Andreas E. Savakis\textsuperscript{b}

\textsuperscript{a} Eastman Kodak Company, Imaging Science Technology Laboratory, Imaging Science Division, Rochester, NY 14650, USA
\textsuperscript{b} Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623, USA

Received 8 October 1999; Revised 3 August 2000; accepted 28 August 2000. Available online 7 August 2001.

Abstract

A two-stage texture segmentation approach is proposed where an initial segmentation map is obtained through unsupervised clustering of multiresolution simultaneous autoregressive (MRSAR) features and is followed by self-supervised classification of wavelet features. The regions of “high confidence” and “low confidence” are identified based on the MRSAR segmentation result using multilevel morphological erosion. The second-stage classifier is trained by the “high-confidence” samples and is used to reclassify only the “low-confidence” pixels. The proposed approach leverages on the advantages of both MRSAR and wavelet features. Experimental results show that the misclassification error can be significantly reduced by using complementary types of texture features.

Author Keywords: Texture segmentation; Complementary features; MRSAR; Wavelet; Confidence map; Boundary refinement; Spatial constraint

---

Article Outline

1. Introduction
2. Initial segmentation using MRSAR coefficients
   2.1. Normalization and weighting of MRSAR coefficients
   2.2. Clustering of MRSAR coefficients
3. Boundary refinement
   3.1. Morphological boundary refinement
   3.2. Refinement using adjustable window for texture feature extraction
   3.3. Self-supervised refinement by reclassification of low-confidence regions based on wavelet features
1. Introduction

Texture is a fundamental characteristic of natural images that, in addition to color, plays
an important role in human visual perception and provides information for image
understanding and scene interpretation. A large volume of research over the past three
decades has addressed the problem of texture segmentation and has produced a number
of review articles and comparative studies [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12].
Unsupervised image segmentation may be defined as the problem of separating image
regions of uniform texture without prior knowledge of the texture types. Texture
segmentation does not pose the same problem as texture differentiation or classification,
where a supervised system is trained to differentiate between known textures. Texture
differentiation is useful in constrained environments where a limited number of textures
is encountered; however, the number of textures in natural images is huge and, in general,
the types of textures present in a particular image are not known a priori.

Much of the texture segmentation work has concentrated on extracting features that are
suitable for texture modeling, followed by feature clustering or classification so that
image regions of uniform texture may be identified [13]. Classic texture features include
those derived from Laws filters [14], co-occurrence matrices (CO) [15 and 16], and
cortex transform modulation functions (CTMF) [17]. More recently, a number of new
texture features have been considered for texture analysis and segmentation, including
multiresolution simultaneous autoregressive (MRSAR) models [18, 19 and 20], Markov
random field (MRF) models [21, 22, 23 and 24] Gabor filters [25 and 26], wavelet
coefficients [27, 28, 29 and 30], and fractal dimension [21 and 31]. The results of
comparing the relative merits of the different types of features have been nonconclusive
and a clear winner has not emerged in all cases [12]. In general, each set of texture
features offers unique advantages but also has limitations compared to other feature
types.

Meanwhile, one common problem in using feature clustering for image segmentation is
that noise in the extracted features may result in misclassification, which takes the form
of holes and other fragments [32]. When trying to minimize the problem due to noisy
features, another interesting yet challenging problem in texture segmentation is
encountered, which may be described as the boundary effect. It usually appears as
inaccurate segmentation of boundaries or superfluous narrow regions at the boundary
between two textures [18 and 32]. It is conjectured that the boundary effect is caused by
misclassification when the trajectory of the feature vectors makes a transition through feature space [26]. To make matters worse, the misclassification may be interpreted as a third texture, depending on the nature of the features for a particular image.

The use of multiple types of features in texture segmentation may be exploited to take full advantage of the strengths of each feature type and alleviate some of the problems, such as the boundary ambiguity encountered when the segmentation is based on a single feature type. Ideally, one would like to select complementary types of features that are not highly correlated and, when combined, can improve the final segmentation result [10]. One method of combining two types of features is to concatenate the feature vectors and perform the classification step on the augmented feature set. This method has been used for combining texture and color features [33] as well as texture features of different types [10]. When this approach is taken, one has to worry about normalization issues and the relative weighting of the features in each feature set, since in most cases the number of parameters in the different feature sets is not equal. In addition, by concatenating all the features into a single vector, the segmentation result is based on a global clustering criterion, even though it may be more suitable to constrain clustering of certain features to a local neighborhood.

A second approach in combining different types of features is to perform a segmentation for each of the feature sets independently, and then combine the segmentation maps into one. The difficulty with this approach is coming up with a selection rule for assigning the appropriate segmentation labels to the final segmentation result, where segmentation maps disagree with each other. The simple rule of assuming that there is a texture boundary in the final segmentation map everywhere that a texture boundary appears in any of the individual segmentation maps does not work well because it results in over-segmentation.

In this paper, a two-stage approach to texture segmentation is proposed where, after the initial segmentation, potential problem regions are identified and reclassified through a second stage of refinement. The first stage of segmentation is based on MRSAR features and a global clustering criterion. The second stage is based on self-supervised clustering of wavelet coefficients applied on the regions of low confidence of the segmentation map. The proposed method can be considered as a novel way of combining different types of texture features. In the next section, the initial segmentation method is described, where an improved way of normalizing MRSAR features is utilized for improved results. In Section 3, a method for extracting the segmentation confidence map using morphological operations is discussed. The segmentation refinement process based on wavelet features is presented in Section 4. Finally, 5 and 6 include experimental results and discussions, respectively.

2. Initial segmentation using MRSAR coefficients

The simultaneous autoregressive (SAR) model for image representation is formulated by
\[ f(i,j) = \sum_{(p,q) \in R} \alpha(p,q) f(i+p,j+q) + w(i,j) \] (1)

where \( f(i,j) \) denotes the intensity of an image pixel at location \((i,j)\), \( \alpha(p,q) \) denotes a weighting coefficient, \( R \) is the neighborhood region, and \( w(i,j) \) is zero-mean Gaussian noise with variance \( \sigma^2 \). Note that the mean value of the image intensity has been subtracted. For a given neighborhood \( R \), the model parameters \( \alpha(p,q) \) and \( \sigma \) may be computed via least-squares estimation (LSE) or maximum likelihood estimation (MLE).

To reduce by half the number of SAR coefficients that need to be computed, a symmetric model is used, where \( \alpha(p,q) = \alpha(q,p) \). The resulting SAR coefficients and noise standard deviation are used as features for texture modeling and segmentation of gray scale images. When processing color images, the SAR coefficients are computed from the intensity or luminance image. To obtain a better estimate of the SAR parameters, the regression is performed over a local window centered at the pixel of interest. The size of the local window should be large enough to allow for accurate and reliable capture of the texture characteristics and small enough to reduce the effects of texture variability over the image.

There are two difficulties in utilizing the SAR model: (a) choosing a proper neighborhood size for the SAR model, within which pixels are considered interdependent, and (b) selecting a proper window size in which the texture is assumed homogeneous and the parameters of the SAR model are estimated. A small neighborhood or window may not be adequate for capturing large-scale textures, while an unnecessarily large neighborhood or window may introduce severe averaging effects, which tend to degrade the discriminatory power of the SAR parameters.

Multiresolution SAR (MRSAR) was proposed as a means to deal with the above problem [18]. If an image with a coarse texture is subsampled, an SAR model with a small neighborhood will fit the subsampled image well, since two neighboring pixels in the subsampled image are several pixels apart in the original image. Therefore, using SAR models at different resolution levels can provide discriminatory power for a variety of texture types.

### 2.1. Normalization and weighting of MRSAR coefficients

The MRSAR coefficients tend to be noisy due to inhomogeneity of the input texture. They also need to be normalized before they are used for segmentation. Additionally, the contrast between features of different texture types may not be optimal for a particular segmentation or classification algorithm. Finally, the discriminatory power of each feature may not be the same. The normalization procedures used in [18] are local averaging, nonlinear transformation, and feature weighting. The procedures used here are (a) local averaging based on a 3x3 local window, (b) a nonlinear transformation that employs a sigmoidal function, as in [18], (c) an improved method of feature weighting that is described below [32], and (d) final feature smoothing based on a 7x7 local window.
Feature weighting is used to emphasize the best features. A feature is considered “good” for segmentation if its within-class variance is small while its between-class variance is large. Mao and Jain [18] assumed that homogeneous textured regions in the images are relatively large in comparison to texture boundaries. Therefore, a good feature should have a small variance within homogeneous textured regions and a large variance over the entire image. For the $k$th feature image, the feature weight is given by

$$w_k = \frac{\text{var}^k_{\text{global}}}{\text{var}^k_{\text{local}}}$$

where

$$\text{var}^k_{\text{global}} = \frac{1}{M^2} \sum_s (f_k(s) - \bar{f}_k)^2.$$  \hspace{1cm} (3)

and

$$\text{var}^k_{\text{local}} = \frac{1}{N^2} \sum_s \left( \frac{1}{P^2} \sum_{s \in N_s} (f_k(s) - \bar{f}_k)^2 \right).$$  \hspace{1cm} (4)

where $\bar{f}_k$ is the mean of the feature over the given $M \times M$ image, and $\bar{f}_k$ is the local mean within a $P \times P$ window, $N_s$, centered at site $s$. All $N$ sites are chosen randomly over the $M \times M$ image. In [18], $P=16$ and $N=64$.

If $N$ is sufficiently large, the measure $w_k$ is greater than or equal to one. Therefore, $w_k$ can be viewed as an estimate of the ratio of between-class variance to within-class variance. However, at texture boundaries, the local variance can become so large that it may skew or inflate the average local variance over the entire feature image, especially when the above-mentioned assumption about the proportion of homogeneous texture regions to texture boundary region is violated. In our previous work [32], we proposed an edge-filtered, nonrandom way of determining the average local variance. In a feature image, if the local variance of a given window is above a predetermined threshold, it does not contribute to the computation of the average local variance, because the large variance is attributed to texture inhomogeneity that is usually found at the boundary between textures. Consequently, the edge-filtered local variance reflects more accurately how homogeneous a feature is within the interior of texture regions. In our experiments, we have found that this measure is better for selecting discriminatory features. The edge-filtered weighting factors are more consistent with the visual interpretation of the discriminating power of the features, while the weighting factors in [18] can sometimes be contrary to visual interpretation.

### 2.2. Clustering of MRSAR coefficients

The clustering of MRSAR coefficients is done via unsupervised segmentation, where the partition of the feature space is purely determined by within-class and between-class Euclidean-like distances, and leads to a simple Voronoi partition. The simple $k$-means algorithm [34] is used to perform the segmentation. For general segmentation
applications not targeted on specific objects or materials, one does not have any specific knowledge of the image content. Moreover, one usually does not even know exactly how many types of materials or objects are in the image. One viable solution to the problem of not knowing the true number of clusters is based on the calculation of the within-cluster distance and the between-class distance. In general, optimal clustering is the one with the smallest within-class distance and the largest between-class distance, where each cluster corresponds to one class. In [35], the within-class distance alone is used to find the true cluster number. Initially the within-cluster distances are high for only one cluster and decrease as the number of clusters approaches the true cluster number. Typically, the within-class distance becomes approximately constant after that. If a ratio of the within-cluster distance and the between-class distance is used, one should expect the ratio to reach a minimum value at the true cluster number. However, in practice it is non-trivial to determine the true cluster number based on the above principles. In the experimental results presented later, the true cluster number is assumed known for simplicity in order to focus on the main contributions of this paper.

3. Boundary refinement

In this section, three alternate methods for texture boundary refinement are described and compared:

(a) boundary refinement using morphological operations;
(b) refinement using adjustable window size; and
(c) self-supervised refinement by wavelet-based reclassification of the low-confidence regions.

It should be noted that the proposed two-stage classification method in (c) may be viewed as a process of combining complementary texture features, as discussed in Section 1.

3.1. Morphological boundary refinement

In texture segmentation, researchers have noticed the appearance of narrow regions due to misclassification, that form a third texture for pixels near the boundary between two textures. Spatial filtering operations have been used to clean up the segmentation maps. In [26], two-step “m-ary” morphological operations are used as postprocessing to reduce such narrow misclassified regions. Note that “m-ary” operations are needed because the segmentation map usually is not simply binary. However, generic gray scale morphological operations are not suitable since they produce intermediate levels that have no meaning for a segmentation map. Therefore, the following operations were defined in [26]:

1. M-erosion: pixels whose neighbors are in the same class are left unchanged, otherwise the pixel is labeled “uncertain”, i.e., there is low confidence in the classification result. Labeling is done by assigning zero value to the “uncertain” pixels while all other segmentation regions retain their original nonzero values.
2. *M-dilation:* “uncertain” pixels are assigned to the most popular class within an 8-neighborhood.

In essence, application of M-erosion followed by M-dilation propagates “certain” regions into the “uncertain” areas based on spatial proximity. It is arguably the most reasonable solution for uncertainty without further a priori information.

### 3.2. Refinement using adjustable window for texture feature extraction

Misclassification often stems from the fact that texture feature extractors are area-based, and a third texture is what the extractor interprets when its window is positioned on two different textures. One method for boundary refinement is to use smaller window sizes near texture boundaries [36]. The boundary refinement process starts by setting $W_b = W$ and utilizes the segmentation map, as obtained from initial clustering, to find regions of thickness $W_b$ around changes in classification. These regions correspond to either boundary areas or small holes. Refinement involves that, for each pixel in the identified regions, texture features are recalculated using a window of size $W_b/2$. Then, the pixel is reclassified according to the distance to locally available cluster centroids, including its original class label and the class labels that fill at least 25% of a window of size $W_b$ centered at the current pixel. The refinement procedure continues by reducing the window size by a factor of two and repeating the above process until a minimum window size is reached, which is sufficiently small to compute useful features; for example, the window size can be as small as $W_b = 5$ [36] for different types of texture features.

Unfortunately, we found that the MRSAR features become rather noisy and unstable once the window size is smaller than $11 \times 11$. We use a $21 \times 21$ window, which is found to be a good compromise among several concerns, and using an increasingly smaller window is not feasible for MRSAR. Another option is to use the highest resolution MRSAR features to reclassify the boundary pixels. However, our experiments show no improvement in the segmentation of boundaries using the highest resolution MRSAR features alone.

### 3.3. Self-supervised refinement by reclassification of low-confidence regions based on wavelet features

Wavelet features are considered here as alternative features for reclassification for several reasons: (1) they are fundamentally different from neighborhood-based MRSAR and MRF types of texture features and capture the texture characteristics in a different way; (2) they are more sensitive to higher resolution variations because of relatively smaller spatial support; and (3) the computational cost is low compared to MRSAR. Gabor filters share a similar spirit of scale-orientation analysis and provide features for more than one orientation. Both Gabor and wavelet features have been used for texture segmentation [26 and 30]. In this study, we perform a three-level wavelet decomposition of the image using Daubechies 9/7 biorthogonal wavelets and construct a nine-dimensional feature vector by properly interpolating the nine high-frequency subbands. An alternative is to use an over-complete wavelet expansion without the need for interpolation. It may also be viable to
use other filterbanks that are more suitable for texture segmentation, which is beyond the focus and scope of this paper.

Before interpolation, the subband coefficients are replaced by their absolute values and are smoothed using a 3×3 local window. In our experiments, we found that the signal-to-noise ratio of wavelet features is not as high as for MRSAR. As a result, the segmentation maps obtained using wavelet features often appear more noisy than those obtained using MRSAR features. Moreover, wavelet features cannot differentiate certain textures that are similar in terms of short correlation (see Fig. 4(d)). On the other hand, MRSAR features are capable of capturing correlation over long distances but often fall short at texture boundaries.

The advantages and disadvantages of MRSAR and wavelet features complement each other and inspired us to devise the two-stage segmentation approach presented here. Note that the initial segmentation map was obtained through unsupervised clustering of the MRSAR features. Therefore, training of the wavelet classifier can be considered as self-supervised according to the “confidence” map. The reclassification is finally based on a minimum distance criterion. In this study, we choose the simple nearest-neighbor classifier for refinement, although a maximum likelihood classifier can be readily trained.

In particular, for each class present in the multilevel confidence map, all the “high-confidence” pixels serve as training samples and are used to determine the corresponding class centroids in the wavelet feature space. The stages for the joint segmentation are illustrated in Fig. 1, where solid arrows indicate process flow and broken arrows indicate information flow. Note that a combination of morphological M-erosion and M-dilation with a small kernel is used in the end to perform minor cleaning of the segmentation map.

Mathematically, the two-stage segmentation approach is formulated as follows. A segmentation of an image $R$ is a finite set of regions $R_1, R_2, \ldots, R_s$, such that

$$R = \bigcap_{i=1}^{s} R_i, \quad R_i \cup R_j = \emptyset, \quad i \neq j. \quad (5)$$

For a pixel $(x, y) \in R$, where $(x, y)$ denotes the spatial coordinates, its initial region label $L(x, y)$ is given by a first classification function $F_1(.)$ based on a first set of MRSAR features $M$, i.e.,

---

Fig. 1. The stages of joint segmentation.

---

Mathematically, the two-stage segmentation approach is formulated as follows. A segmentation of an image $R$ is a finite set of regions $R_1, R_2, \ldots, R_s$, such that

$$R = \bigcap_{i=1}^{s} R_i, \quad R_i \cup R_j = \emptyset, \quad i \neq j. \quad (5)$$

For a pixel $(x, y) \in R$, where $(x, y)$ denotes the spatial coordinates, its initial region label $L(x, y)$ is given by a first classification function $F_1(.)$ based on a first set of MRSAR features $M$, i.e.,
\[ L(x,y)=i, \text{ if } F_1[\mathbf{M}(x,y)]=i, \quad i \in \{1, 2, \ldots, s\}. \]  

(6)

Let the boundary of \( R_i \) be denoted by \( \partial R_i \), then the set of “low confidence” pixels, \( R_L \), is given by

\[ R_L = \bigcup_{i=1}^{s} \partial R_i, \]  

(7)

and the set of “high confidence” pixels, \( R_H \), is the collection of interior pixels

\[ R_L = \bigcup_{i=1}^{s} (R_i - \partial R_i). \]  

(8)

Note that \( R=R_H \cup R_L \), \( R_H \cap R_L = \emptyset \).

In the second stage, a “low confidence” pixel \( (x,y) \in R_L \) is reclassified according to a second classifier \( F_2(.) \) according to a second set of wavelet features \( \mathbf{W} \)

\[ L(x,y)=j, \text{ if } F_2[\mathbf{W}(x,y)]=j, \quad j \in \{j_1, j_2, \ldots, j_{t(x,y)}\}, \mathcal{V}(x,y) \in R_L. \]  

(9)

The second classifier \( F_2(.) \) is determined based on the second set of wavelet features \( \mathbf{W} \) collected at all the training sample pixels in \( R_H \). It is noteworthy that instead of the entire set of all labels, a local set of \( t \) region labels \( \{j_1, j_2, \ldots, j_{t(x,y)}\} \) are identified using the adaptive neighborhood window around the pixel of concern at \( (x,y) \), as will be described next.

### 3.4. Determination of spatially adaptive neighborhood for reclassification

A neighborhood window is used in the reclassification of uncertain pixels to guarantee that holes and narrow strips do not exist in the final segmentation. In the light of the discussion in Section 1, this neighborhood window serves as an additional constraint to limit the classification errors that would be committed by the second set of texture features, especially at the locations where there is high confidence in the classification by the first set of features. In Fig. 2(a), the adaptive neighborhood window for each identified “uncertain” pixel is determined by

1. Starting at the pixel itself (the solid black pixel).
2. Attempting to expand a rectangular window by one pixel in each of the four directions (up, down, left, right, which form an expanding cycle) until at least one of the following is true: (a) an image border is reached by the rectangle; (b) at least one of the recently expanded rectangle pixels (the white pixels), excluding corner pixels of the current rectangle (the shaded pixels, since they are on two sides of the rectangle at the same time), is a labeled pixel; and (c) the maximum search range is reached in a given direction.
3. Continuing the overall expansion until one of the following is true: (a) for a noncorner pixel, expansion is not possible or necessary for at least a pair of opposite directions; and (b) for a corner pixel, expansion is not possible or necessary in all directions.

![Diagram](6K)

Fig. 2. Determination of spatially adaptive neighborhood for reclassification. In the example cases, the current uncertain pixel is indicated by the small open circle and the neighboring regions are indicated by $R_i$.

A few typical cases are shown in Fig. 2: (b) boundary, (c) intersection, (d) hole, and (e) corner. It is clear that such a spatially adaptive neighborhood window ensures that a uncertain pixel is always reclassified to a locally available class. In particular, the requirement of searching for candidate classes in at least a pair of opposite directions, ensures that all of the *immediately* available classes, rather than only the *closest* class, are used. Using the closest class would produce the same effect as morphological M-dilation due to the sole use of spatial proximity.

### 4. Experimental results

The results are based on texture mosaic images composed of patterns from the Brodatz set [37] and the VisTex natural scene collection by MIT, where the size of each uniform texture patch is 128×128. The uncertainty maps are created by “$m$-ary” morphological erosion operations. The segmentation results by either MRSAR or wavelet features are obtained using a $k$-means clustering algorithm. In general, the segmentation maps by wavelet features are more noisy and contain more fragments. The final segmentation is obtained from morphologically cleaned MRSAR segmentation and from the two-stage segmentation/refinement using MRSAR and wavelet features, respectively. Segmentation error maps are also shown for a visual comparison of the results obtained from the two methods.

The first set of images are composites of Brodatz textures. Fig. 3 shows the results for a four-patch mosaic with horizontal and vertical boundaries. Fig. 4 shows the results for a five-patch mosaic consisting of circular and diagonal boundaries. The algorithms are also tested on an eight-patch mosaic in Fig. 5. These experimental results have shown the efficacy of the proposed segmentation approach in generating significantly more accurate segmentation than using each set of features alone for a variety of textures. Moreover, the improvement is superior to morphological postprocessing that is solely based on spatial proximity. In many cases, the error in MRSAR segmentation appears as a boundary shift due to perhaps the uncertainty associated with large windows (texture is inherently an
area-based process). Clearly, such a boundary shift cannot be resolved by using morphological operations alone.

Fig. 3. Four-patch Brodatz texture mosaic. Results in Fig. 3, Fig. 4 and Fig. 5 are annotated using the following convention: (a) original image, (b) uncertainty map (confidence map), (c) segmentation by MRSAR features, (d) segmentation by wavelet features, (e) refined MRSAR segmentation by morphological operations, (f) joint segmentation, (g) error in morphological refinement, and (h) error in joint segmentation.

Fig. 4. Five-patch Brodatz texture mosaic.

Fig. 5. Eight-patch Brodatz texture mosaic.

The self-supervised reclassification proves to be effective, as shown by Fig. 4. Although wavelet features were not able to resolve two similar textures under unsupervised $k$-means clustering the textures are separated by self-supervised reclassification, with the help of spatial constraints. The failure of the wavelet feature-based segmentation does not seem to be a result of poor initialization of the cluster centroids in the $k$-means clustering process. The result shown in Fig. 4 is a typical result out of ten trials.
The second set of images are composed of texture patches extracted from a VisTex collection of natural scenes. Comparable performance was demonstrated in both Fig. 6 and Fig. 7. The result by the two-stage approach would have been better if the wavelet features were better at discriminating the particular concrete texture in Fig. 6 (the illumination change in the concrete texture may also confound the problem).

Fig. 6. Four-patch V is Tex texture mosaic (1).

Fig. 7. Four-patch V is Tex texture mosaic (2).

On the average, based on ground truth, the misclassification rate is reduced by two-thirds compared to the direct MRSAR-based segmentation, and by half compared to morphological cleaning operations, respectively. Furthermore, it seems that the errors are more symmetrically distributed around the true texture boundaries using the two-stage approach, giving the appearance of straighter boundaries. In comparison, morphological cleaning of MRSAR segmentation was unable to correct the boundary shift.

It is noteworthy that the computational cost of the proposed two-stage algorithm is only slightly higher than the MRSAR-based segmentation algorithm. In general, Daubechies 9/7 filter-based wavelet feature extraction take less than 1% of the time needed by MRSAR feature extraction. Even with the additional pre-processing and post-processing involved, the proposed two-stage algorithm takes about 7% more time. It is reasonable to expect such a trend for complementary types of texture features other than MRSAR and wavelet, given the requirements on texture characterization properties of the two types of features, which are summarized in the next section.

5. Discussion and conclusions
A two-stage approach to texture segmentation was proposed in this paper, for taking advantage of the relative strengths in different feature types. The two-stage scheme presented in this paper can be generalized beyond MRSAR and wavelet features. The criterion for selecting two sets of features could be based on the following rule: the first set of features that is used for the initial segmentation should have high signal-to-noise ratio in the interior of homogeneous textured regions while the second set of features should have high spatial resolution. A single set of features is unlikely to have both properties due to the fact that texture is an area-oriented characteristic and texture features are computed within a local neighborhood. Features based on larger local neighborhood windows capture longer correlation and are more robust, however, they suffer from poor spatial resolution near boundaries between textures. On the other hand, features based on smaller local neighborhood windows tend to have better spatial resolution, but may be noisy and sensitive to subtle textural differences. The proposed two-stage segmentation scheme is designed to play to the strength of two sufficiently different sets of texture features while overcoming their drawbacks. Combined with an efficient way of identifying “uncertain” regions where misclassification is likely to occur, and a self-supervised training mechanism, it offers an alternative superior to postprocessing by morphological operations alone.

In summary, we can characterize the advantages and disadvantages of three ways of integrating different types of texture features as follows. The first way of forming a conglomerate feature is straightforward but is likely to suffer from global parameters such as the weighting of different features and finds it difficult to switch to a locally optimal subset of features across the image. With the second approach, segments obtained by different sets of features need to be merged together in a coherent fashion. Again, each segmentation process is based on a global setting of parameters so that it is not difficult to form optimal decision boundaries in the feature space for a local area in the image. It is unclear how to derive an appropriate label when different segmentation maps disagree with each other. The final result is also susceptible to fragmentation due to small misalignments between the segmentation boundaries drawn using different features because of lack of coordination between two essentially separate segmentation processes. The third way of using a two-stage scheme designed according to the strength and weakness of two sets of complementary features seems to provide a superior alternative. To a large extent, the function of the second stage of classification is reduced to a two-class decision, restricted within an identified local area, supervised by the high-confidence samples, and according to a set of features with higher spatial resolution. In addition, further misclassification is limited to locally present texture classes. The two-class decision boundaries are redefined in a more appropriate feature space suitable for resolving local classification ambiguities in any given spatially confined area. Therefore, a greater degree of flexibility and sophistication is provided to accommodate changing textural characteristics across the image.

6. Summary

A number of different features have been used for texture segmentation in the past, however, no single type of feature has emerged as a clear winner in all cases.
Complementary types of features, when carefully chosen, may be used jointly to improve the segmentation results obtained by using any single one of the feature types. In particular, multiresolution simultaneous autoregressive (MRSAR) and wavelet features may be viewed as complementary, in the sense that MRSAR features have large support and provide good texture discrimination, yet suffer from misclassification that usually occurs near the texture boundaries, while wavelet features offer good localization at the expense of lower signal-to-noise ratio. In this paper, a two-stage texture segmentation approach is proposed where an initial segmentation map is obtained through unsupervised clustering of MRSAR features and is followed by self-supervised classification of wavelet features. The self-supervised stage is based on a segmentation confidence map, where the regions of “high confidence” and “low confidence” are identified on the MRSAR segmentation result using multilevel morphological erosion. The second-stage wavelet classifier is trained from the “high-confidence” samples and is used to reclassify only the “low-confidence” pixels. The final reclassification is based on rules that combine minimum distance and spatial constraints. Additionally, an improved coefficient feature normalization procedure is used during the classification process of both stages. The proposed two-stage approach leverages on the advantages of both MRSAR and wavelet features, and incorporates an adaptive neighborhood-based spatial constraint. Experimental results show that the misclassification error can be significantly reduced compared to morphological cleaning operations alone, at a slightly increased overall computational cost.

References


Corresponding author. Tel.: 716-722-7139; fax: 716-722-0160

Vitae

**Luo**: About the Author—JIEBO LUO received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1989 and 1992, respectively. In 1995, he received the Ph.D. degree in electrical engineering from the University of Rochester, Rochester, NY. He was a Research Assistant in the Image Processing Laboratory and the NSF Center for Electronic Imaging
Systems at the University of Rochester, from September 1992 to November 1995. In the summer of 1995, he was employed at the Joseph C. Wilson Center for Technology of Xerox Corporation, Webster, NY. In 1995, he became a Senior Research Scientist, and is currently a Research Associate in the Imaging Science Technology Laboratory, Imaging Research and Advanced Development, Eastman Kodak Company, Rochester, NY. His research interests include image enhancement and manipulation, digital photography, image and video coding, wavelets and applications, medical imaging, pattern recognition, and computer vision. He has authored over 50 technical papers, and over 10 issued or pending US and European patents. Dr. Luo was the recipient of the Best Student Paper Award for Visual Communication and Image Processing from SPIE in 1994, and a Certificate of Merit for Scientific Exhibit from RSNA in 1998. He has been serving the Rochester Section of the IEEE Signal Processing Society as an officer for the past two years, and was the General Co-Chair of the 2000 IEEE Western New York Workshop on Image Processing. Dr. Luo is a Senior Member of the IEEE and a member of SPIE.

**Savakis: About the Author**—ANDREAS SAVAKIS received the B.S. (Summa Cum Laude) and the M.S. degrees in Electrical Engineering from Old Dominion University in 1984 and 1986 respectively, and the Ph.D. in Electrical Engineering from North Carolina State University in 1991. From 1991 to 1996 he was a Research Scientist at the University of Rochester where he conducted research in the areas of pattern recognition, infrared imaging, digital radiography, and human vision. In 1996, he joined the Eastman Kodak Company and worked at the Business Imaging Division and the Research Laboratories. His research at Kodak involved algorithm development for high-speed scanners, image understanding and multimedia applications. Since Spring 2000 he is Associate Professor at the Computer Engineering Department of the Rochester Institute of Technology, where he currently serves as the department head. His current research interests include image segmentation, scene classification and hardware implementation of imaging algorithms. Dr. Savakis is a Senior Member of the IEEE and member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.